☐   Generate Collection | Print

L10: Entry 15 of 17   File: USPT   Feb 22, 2000

DOCUMENT-IDENTIFIER: US 6028962 A
TITLE: System and method for variable encoding based on image content

Abstract Text (1):
A system and method for efficient image encoding begins by loading a series of image blocks. A current block is compared with an immediately preceding image block for coincidence, and coded as matching or nonmatching based on the comparison results. The system further compares the nonmatching image blocks with a stack containing a list of the most recently encountered image blocks. Image blocks matching an entry in the list are coded by reference to the appropriate entry in the list. In the event a match is still unavailing, the system determines if the image blocks may be categorized as a bilevel text block, a bilevel image block, a block containing only one gray pixel, or a block containing multiple gray pixels. Each nonmatching block is classified according this classification criteria.

Brief Summary Text (5):
Data compression is commonly used in digital systems to reduce the quantity of bits required to represent digital data. Specifically, data compression may be employed to store digital data using less memory space, or to transmit digital data more quickly. Data compression is especially prevalent in the processing of digital images due the large volume of information involved.

Brief Summary Text (10):
In most instances, the above identified compression techniques are specifically tailored to process a particular type of image, such as a bilevel image. And in fact, these compression techniques perform well so long as they are fed images which they were designed to handle. Yet significant problems occur when a data compression technique designed to process a particular type of image is fed a "foreign" type of image. This problem may arise, for example, when a raw image document contains a mixture of different types of images.

Brief Summary Text (21):
In another embodiment, the above described classification may be embedded in a larger compression algorithm which uses a combination of Huffman encoding and stack-based tables. According to this exemplary embodiment, the encoding scheme begins by successively stepping an M.times.N window through a buffer of image data. As the window is moved through the image, the contents of a current window are compared to the contents of an immediately preceding window. More particularly, the window is divided into quadrants, and each quadrant of the current window is compared with its counterpart quadrant in the previous window. Based on the comparison, a Quad Match bit is set to reflect whether a match has occurred. If a Quad Match bit indicates a match has occurred, the corresponding quadrant pixels do not have to be included in the transmitted data stream.

Brief Summary Text (23):
If the above comparison fails to produce a match, the quadrants are then examined according to the above described classification scheme to discriminate whether it is a bilevel text window, bilevel image window, one gray level window or multiple gray level window.

Drawing Description Text (5):
FIGS. 3(a), 3(b) and 3(c) illustrate the basis for comparison between blocks of data using step rates of 6, 8 and 10 pixels per step, respectively.

Drawing Description Text (6):
FIG. 4 provides a truth table outlining the manner of changing a step rate in response to comparison of image blocks at step rates of 6, 8 and 10 pixels per step.

Detailed Description Text (2):
The present encoding system and method may generally be employed to efficiently code images for transmission or storage. By way of example, the encoding scheme may be used to code images for efficient transmission to a printer, facsimile, or simply for database file storage. However, to facilitate discussion, the present invention will be explained within the context of printer data compression.

Detailed Description Text (6):
As shown in FIG. 1, the compression unit (2) is connected to a system bus (4) via bus interface (6). In an exemplary embodiment, the compression unit functions as a slave processor, and compresses data when so commanded by the host system.

Detailed Description Text (12):
Once the windows are loaded, the matching and encoding module (12) comes into play by first checking for an exact match between pixels in a current window (24) and pixels in the window which immediately preceded the current window--referred to as the previous window (22). Often, printed data will exhibit a repetitious nature depending on the nature of the font used to generate the text or the half-tone matrix that was used to render the image. Accordingly, the current and previous windows are compared using the different step rates (6, 8 and 10 pixels) in an attempt to identify this natural repetition. FIGS. 3(a),(b) and (c) illustrate the specific basis for comparison using step rates of 6, 8 and 10, respectively.

Detailed Description Text (13):
The step rate of the window loading logic may initially be set at 8 pixels per step. If the above comparison step indicates that this step rate is out of sync with the natural cycle of data in the strip, the step rate is changed. FIG. 4 provides a truth table indicating the specific manner in which the step rate is changed. For instance, in the case where the step rate is initially set at 8, a match at window length of 8 will require no change in the step rate. However, if a match occurs at a window length of 6 pixels, then the table instructs the compression unit (2) to switch to a lower step rate of 6. Similarly, if a match occurs at a window length of 10 pixels, then the table instructs the system to switch to a higher step rate of 10. If none of the step rates produces a match, then the table instructs the system to maintain the current step rate. Furthermore, if there is a match for a specific window which comprises all white or all black pixels (denoted in the table as CV* for "Constant Value"), the table instructs the system to maintain the current step rate. More specifically, in the event of a constant value window, a change in step rate is inhibited to allow a more consistent lock on the half-tone frequency.

Detailed Description Text (16):
In general, the matching and encoding module (12) employs three principal hierarchical phases in coding the quadrants. First, the module (12) compares a current quadrant from the current window with the corresponding quadrant from the previous window. If a match is found, pixel values comprising the current quadrant do not have to be included in the output data stream.

Detailed Description Text (17):
If a match is unavailing, however, the encoding module (12) enters the second phase of its examination. In the second phase, the unmatched current quadrant is compared with a stored list of previously encountered image quadrants, starting from the second-to-last recently encountered image quadrant (the immediately preceding image quadrant having already been checked, as described above). If a match is found between the current quadrant, and an entry on the list, then the quadrant is coded by making reference to the entry on the list.

Detailed Description Text (21):
As part of the first phase, the matching and encoding module ((12 in FIG. 1)) compares a quadrant taken from the current window with its respective counterpart quadrant taken from the previous window (step S4 in FIG. 5). If they match, a Quad Match bit corresponding to that

quadrant is set to 1 (Step S6). Otherwise, this bit is set to 0 (Step S8). This procedure is repeated for the quadrants to produce a series of Quad Match bits. Only the image data within the quadrants assigned a Quad Match bit of 0 need to be transmitted in the output data stream. Furthermore, if the series of Quad Match bits duplicates a previous series of Quad Match bits, these bits do not have to be transmitted.

Detailed Description Text (27):
In attempt to further compress the current window, the unmatched quadrants are compared with a stack (14) containing a list of most recently used image data (step S10 of FIG. 5). The stack (14) is specifically comprised of four different stacks, each holding data corresponding to quadrants 1-4, respectively. As illustrated in FIG. 7, each of the four stacks consists of 16 elements storing data (28) corresponding to the last 16 image quadrants encountered. Any item in the stack can be promoted to the most recently used position--or top--of the stack. Any new item inserted into the stack causes the least recently used item (LRU) to conceptually fall off the bottom of the stack and all other items to shift down one position. In this manner, each quadrant stack eventually contains the 16 most recently used quad bit values for that specific quadrant in order from most recently used (MRU) to least recently used (LRU).

Detailed Description Text (28):
As readily understood by those skilled in the art, the stack data is not actually shifted in response to promotion or demotion of entries in the stack. Rather, pointers (30) to the stack entries are manipulated to indicate the ordering of the stack. For instance, an item may be promoted from any level within the stack to the MRU by loading the pointer from that level, Tn, into the T0 slot, and shifting all the pointers from T0 to Tn-1 down one level. Inserting a new item into the stack is essentially the same as promoting the pointer in T15 to the top and storing the new item's data at that pointer's location.

Detailed Description Text (55):
The present invention with its novel encoding scheme has been described in full within the exemplary context of printer data compression. However, as discussed above, the present invention is not limited to encoding printable data.

CLAIMS:

4. A system for encoding a digital image as claimed in claim 1, wherein said encoding module classifies said grouping as a singular gray level grouping and encodes said grouping to indicate said classification when said type of data contained within said grouping contains only one gray level datum.

6. A system for encoding a digital image as claimed in claim 1, wherein said encoding module classifies said grouping as a multiple gray level grouping and encodes said grouping to indicate said classification when said type of data contained within said grouping contains more than one gray level datum.

7. A method for encoding a digital image as claimed in claim 1, wherein said encoding step classifies said grouping as a multiple gray level grouping and encodes said grouping to indicate said classification when said type of data contained within said grouping contains more than one gray level datum.

11. A method for encoding a digital image as claimed in claim 8, wherein said encoding step classifies said grouping as a singular gray level grouping and encodes said grouping to indicate said classification when said type of data contained within said grouping contains only one gray level datum.

13. A system for encoding a digital image including:

a loading module for identifying a current image grouping within said digital image;

a matching and encoding module for comparing said current image grouping with a previous image grouping, and coding said current image grouping as a matched image grouping when the current

image grouping matches a previous image grouping, else coding said current image grouping as a nonmatched image grouping;

said matching and encoding module further identifying the type of data contained within said nonmatched image grouping as either bilevel data or multilevel data, and providing coding for said nonmatched image grouping on the basis of the type of data so identified;

an output module for outputting the encoded image grouping.

16. A system for encoding a digital image as claimed in claim 13, wherein said system further includes:

a stack memory containing a list of previous groupings processed by the system;

and wherein said matching and encoding module further compares said nonmatched grouping with entries contained in said list.

20. A method for encoding a digital image including:

identifying a current image grouping within said digital image;

comparing said current image grouping with a previous image grouping;

coding said current image grouping as a matched image grouping when the current image grouping matches a previous image grouping, else coding said current image grouping as a nonmatched image grouping;

identifying the type of data contained within said nonmatched image grouping as either bilevel data or multilevel data;

providing coding for said nonmatched image grouping on the basis of the type of data so identified;

outputting the encoded image grouping.

23. A method for encoding a digital image as claimed in claim 20, further comprising the steps of:

comparing said nonmatched grouping with a stack containing a list of prior groupings; and

coding said nonmatched grouping by reference to a corresponding item in said list, if it is determined that said nonmatched grouping corresponds with said item.

27. A method for encoding image data, comprising the steps of:

storing data in a stack which identifies values for plural groups of pixels in an image;

selecting a group of pixels within the image;

comparing pixel values for said selected group of pixels with data stored in said stack to determine whether the pixel values for said selected group match data for any group of pixels which is stored in said stack;

encoding the pixel values for said selected group as matching data stored in said stack when there is a match;

storing data which identifies the pixel values for the selected group in said stack when there is no match;

encoding the pixel values for the selected group as new pixel value data when there is no

match; and

reordering the data stored in said stack so that the plural groups are listed in order from the most recently selected group to the least recently selected group.

Previous Doc   Next Doc   Go to Doc#

Freeform Search

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

L9 and (data near item)

Display: 100 Documents in **Display Format:** FRO Starting with Number 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search   Clear   Interrupt

Search History

DATE: Friday, January 20, 2006   Printable Copy   Create Case

Set Name	Query	Hit Count	Set Name
side by side			result set

DB=USPT; PLUR=YES; OP=OR

Set Name	Query	Hit Count	Set Name
L10	L9 and (data near item)	17	L10
L9	L8 and compar$	77	L9
L8	L7 and (classification near type)	87	L8
L7	L4 and (classif$ near type)	260	L7
L6	L5 and (classif$ near type)	0	L6
L5	L4 and (compar$ near data near item)	12	L5
L4	compress$ near data	26082	L4
L3	L1 and classification	1	L3
L2	L1 classification	63409	L2
L1	5864860.pn.	1	L1

END OF SEARCH HISTORY

Hit List

Search Results - Record(s) 1 through 2 of 2 returned.

☑ 1. Document ID: US 5864860 A

Using default format because multiple data bases are involved.

L1: Entry 1 of 2   File: USPT   Jan 26, 1999

US-PAT-NO: 5864860
DOCUMENT-IDENTIFIER: US 5864860 A

TITLE: Compression of structured data

DATE-ISSUED: January 26, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Holmes; Keith	Waterloo			BE

US-CL-CURRENT: 707/101; 341/51, 707/100, 710/68

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: DE 69718085 E, GB 2310055 A, EP 789309 A2, EP 789309 A3, KR 97062942 A, US 5864860 A, KR 233355 B1, EP 789309 B1

L1: Entry 2 of 2   File: DWPI   Feb 6, 2003

DERWENT-ACC-NO: 1997-375500
DERWENT-WEEK: 200318
COPYRIGHT 2006 DERWENT INFORMATION LTD

TITLE: Multi-record structured data compression - comparing data item in current field with
data item in corresp field of preceding record and if data matches that in corresp field,
replacing it by match indicating token

INVENTOR: HOLMES, K; HOLMES, K T

PRIORITY-DATA: 1996GB-0002550 (February 8, 1996)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
DE 69718085 E	February 6, 2003		000	G06F017/30
GB 2310055 A	August 13, 1997		021	G06F017/30
EP 789309 A2	August 13, 1997	E	012	G06F017/30
EP 789309 A3	November 26, 1997		000	G06F017/30

KR 97062942 A	September 12, 1997		000	G06F017/30
US 5864860 A	January 26, 1999		000	G06F017/30
KR 233355 B1	December 1, 1999		000	G06F017/30
EP 789309 B1	January 2, 2003	E	000	G06F017/30

INT-CL (IPC): G06 F 17/30; G06 F 17/40

ABSTRACTED-PUB-NO: GB 2310055A
BASIC-ABSTRACT:

The method involves for a current field within a current record other than the first record in
the data sequence comparing (250) a data item in the current field with the data item in the
corresp field of a preceding record. If the data item matches the data item in the corresp
field, it requires replacing (270) the current field data item by a token indicating the match
and repeating procedure for a set number of fields in a number of records of the data sequence.
That comparison is repeated for the set number of fields in every record of the data sequence
e.g. for every field in the current record.

The data item in the current field is compared with the data item in the corresponding field of
the immediately preceding record, and the method further comprises, subsequent to performing
the comparison step for the set number of fields in the current record. The uncompressed
current record stored (290) as the immediately preceding record for use in the performance of
the comparison on the set number of fields of a subsequent record.

ADVANTAGE - Alleviates cost of maintaining and replicating structured data.
ABSTRACTED-PUB-NO:

US 5864860A EQUIVALENT-ABSTRACTS:

The method involves for a current field within a current record other than the first record in
the data sequence comparing (250) a data item in the current field with the data item in the
corresp field of a preceding record. If the data item matches the data item in the corresp
field, it requires replacing (270) the current field data item by a token indicating the match
and repeating procedure for a set number of fields in a number of records of the data sequence.
That comparison is repeated for the set number of fields in every record of the data sequence
e.g. for every field in the current record.

The data item in the current field is compared with the data item in the corresponding field of
the immediately preceding record, and the method further comprises, subsequent to performing
the comparison step for the set number of fields in the current record. The uncompressed
current record stored (290) as the immediately preceding record for use in the performance of
the comparison on the set number of fields of a subsequent record.

ADVANTAGE - Alleviates cost of maintaining and replicating structured data.

| Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw Desc | Clip Img | Ima |

| Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS |

Term	Documents
"5864860"	2
5864860S	0
"5864860".PN..PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	2

(5864860.PN.).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.   2

Display Format: |- | Change Format |

Previous Page   Next Page   Go to Doc#

☑   Generate Collection   Print

L9: Entry 2 of 2   File: USPT   Sep 3, 2002

DOCUMENT-IDENTIFIER: US 6446068 B1
TITLE: System and method of finding near neighbors in large metric space databases

Detailed Description Text (25):
Once the p nearest neighbors of the current item X[i] are found in block 96, the near neighbor
links for the current item are stored 98. They may be stored in an array variable containing
all links for the database, if a pointer to an item's links is stored with the item. The links
from an item may be (as preferred) stored near the linking item's description and/or near other
information related to the linking item. In any case, it is desirable that the links from an
item can be located relatively easily, given a reference to the item.

Detailed Description Text (38):
Note that it is possible, given the link structure, that an item in the current search set will
have a link to another item which was or is already in the current search set. This could
result in computing the same query-to-item distance multiple times. This may be easily avoided
by maintaining, during each search, a list of all items which have been compared to the query,
along with the corresponding distances. Well-known hash table techniques can allow fast access
to items in such a list. If a linked item is already in this list, its distance to the query
may simply be read from the list rather than re-computed.

L10: Entry 2 of 3   File: USPT   Aug 26, 2003

DOCUMENT-IDENTIFIER: US 6611842 B1
TITLE: Method and apparatus for generating profile data

Brief Summary Text (11):
There is therefore a need for an alternative method of establishing categorization data
classifying individuals personal preferences for products utilizing purchase and rental or
similar user selection histories which more accurately classifies the personal preferences of
individuals. In particular, there is a need for generating and recording data indicative of a
more accurate classification of personal preferences in a way which can be utilized to select
further products or services based on the underlying core values of preferences held by an
individual.

Detailed Description Text (7):
By identifying for each product within user history data, products associated with high and low
values for classification data and then determining for the entire product history data the
number of occurrences of extreme values within a category or occurrences of groups of extreme
values data indicative of the personal preferences of that user can be generated.

Detailed Description Text (8):
In particular, by generating data indicative of the presence both high category value data and
low category value data associated with products identified within user history data, the
generated classification data identifies mutually inconsistent preferences or underlying values
relied upon at different times. Thus, for example, if the products within user history data are
films which were associated with data indicating that they were extremely fast paced or
extremely slow paced, by identifying the occurrence of these extreme values the individual
personal preferences would be classified in terms of two mutually inconsistent underlying
criteria relied upon on different occasions.

Detailed Description Text (17):
The processing of the control module program 10 to generate data indicative of a classification
of a user's personal preferences will now be described with reference to FIGS. 4 to 6.

Detailed Description Text (47):
The control module 10 then (S10) outputs as a classification of a user a predetermined number
of categories, category pairs and category triples which associated with the greatest values
within the profile data 40,42,44 stored within the profile storage module 16. This data is then
indicative of a classification of the users preferences in terms of the underlying values
associated with the products previously selected by that user.

Detailed Description Text (58):
When the control module 10 determines (S54) that product data has been stored within the user
suggestions storage module 18 for search data corresponding to the last category or groups of
categories in the generated profile the control module 10 then (S58) filters the generated list
by comparing each item in the lists stored within the user suggestion storage module 18 with
the list of products 22 associated with the input user identification data 20 used to generate
data indicative of a user selection strategy and removing from the list stored within the user
suggestion storage module 18 any products corresponding to products within the list 22 for the
record from the user selection database 18. Thus in this way the list generated is filtered to
remove references to products which have already been obtained by that user.

Detailed Description Text (59):
The control module 10 then identifies within the list any items of product data which appear
more than once within the list and deletes these duplicates from the list. The control module
10 then causes the filtered list to be displayed (S60) on the display 7 thereby showing to a
user a list of products identified as corresponding to the classification of a user's personal
preferences which the user has not previously purchased or rented.

CLAIMS:

4. The computer system according to claim 1, further comprising a selection means for utilizing
the user profile data to identify an output item having classification data corresponding to
the user profile data.

9. The method according to claim 6, further comprising utilizing the user profile data to
identify an output item having classification data corresponding to the user profile data.

13. The computer apparatus of claim 12, wherein: said association means comprises database
means for storing a plurality of records associating a plurality of selections each with
classification data comprising data indicative of an assessment of the content of said
selection with respect to a plurality of categories; and determination means for determining
and associating selections with categories corresponding to predetermined ranges indicative of
high or low evaluation by said records associated with said selections.

15. The computer apparatus of claim 14, wherein said selection means comprises search data
generation means for generating search data utilizing said user profile data, and selection
output means for outputting data associated with classification data within said database means
associated with classification data substantially corresponding to said search data generated
by said search data generation means.

16. The computer apparatus of claim 15, wherein said selection means is arranged to determine a
correspondence between said search data and said classification data base on a determination of
a value indicative of a sum of absolute values of differences between said search data for a
category and said classification data for the same category for at least some of said plurality
of categories.

18. A method in accordance with claim 17, wherein said association step comprises the steps of:
storing a plurality of records associating a plurality of selections each with classifications
data comprising data indicative of an assessment of the content of said selection in respect of
a plurality of categories; and determining and associating selections with categories
corresponding to predetermined ranges indicative of high or low evaluation by said records
associated with said selections.

20. A method in accordance with claim 19, wherein said selecting step comprises the steps of
generating search data utilizing said user profile data, and outputting data associated with
classification data substantially corresponding to said generated search data.

21. A method in accordance with claim 20, wherein said outputting-step comprises determining a
correspondence between said search data and said classification data based on a determination
of a value indicative of a sum of absolute values of differences between said search data for a
category and said classification data for the same category for at least some of said plurality
of categories.

☐ ▓▓▓ Generate Collection ▓▓▓ | Print

L11: Entry 8 of 8   File: USPT   Apr 13, 1999

DOCUMENT-IDENTIFIER: US 5893717 A
TITLE: Computerized method and system for teaching prose, document and quantitative literacy

Brief Summary Text (19):
According to Kirsch and Mosenthal, as set forth in the articles, substantially all documents can be classified into a reasonably limited number of types, and each document type has a substantially limited number of uses. The document types are: (a) the four kinds of matrix documents: simple lists, combined lists, intersecting lists, and nested lists; (b) the three kinds of graphic documents: pie charts, bar graphs, and line graphs; (c) entry documents or forms; (d) the two kinds of maps: general reference maps and thematic maps; and (e) the four kinds of mimetic documents: pictures, diagrams, process schematics, and procedural schematics. Each of these document types will be discussed briefly in turn.

Brief Summary Text (40):
This instructional limitation is further compounded by the lack of consistency, in current document pedagogy, for labeling and classifying different document structures, contents, and strategies. In many cases, the nomenclature of documents in various instructional practices varies from one approach to another. For instance, in some instructional approaches, a "chart" is defined as a "map" and a "graph" is defined as a "diagram." In turn, a "diagram" is defined as a "schematic" and a "schematic" is defined as a "picture." As instructional designers have demonstrated, without a consistent framework by which teachers, as well as students, can consciously understand and discuss the dimensions of a problem, students have no basis for solving document-related problems effectively. In terms of completing document tasks, this means that, without a common rhetoric for identifying and describing the underlying structures, contents, and strategies of documents, teachers have few universal tools for equipping themselves and their students to understand and master Document Literacy Skills.

Brief Summary Text (66):
This menu-driven nature of the Computerized Literacy System accomplishes several objectives. One objective, in accordance with the Knowledge Model Procedure, is to teach students of the limited number of document types into which all documents can be classified. A second objective is to keep track of what a given set of students has learned. A third objective is to permit the users to receive instruction about the document type which is most important or relevant to them. A fourth objective is simply to make the system easy to operate, even for users with little or no familiarity with computers.

Detailed Description Text (26):
The structure lessons for simple lists focus on the key concept of related information. Students are taught that to make up a simple list, items must have something in common. They also are taught that the function of labels is to define the common characteristic shared by a group of items. This concept of relatedness forms the basis for classifying and categorizing information. If students understand how information is put together in this simplest structure, they will be in a much better position to understand increasingly complicated document structures as they progress through the system. The lesson segment menu in one embodiment is in FIG. 7.

Detailed Description Text (43):
In the third lesson segment, "More About Labels", the function of labels is explored. Students

are taught that labels show what a group of items has in common. They are also taught that when lists do not have labels, <u>classifying</u> or categorizing the items is the way to supply the missing label.

<u>Detailed Description Text</u> (168):
The Lesson Segment. Students are first told that, One common way to use combined <u>lists is to compare items in the lists</u>. There are clue words in questions that tell you to compare. The clue words "more," "less," "most," "least" and the endings -er and -est appear on the video monitor. Students can then select from three boxes on the screen: amounts, prices, and times. Clicking on a box displays a combined list and a comparing question to be discussed. NEXT QUESTION shows an additional comparing question for the document. NEXT EXAMPLE displays another combined list in the category selected. There are two documents for each category and two questions for each document.

<u>Detailed Description Text</u> (640):
Students are then shown a screen display depicting a schematic of the growth of a seed into a plant, and are told that the text contains major steps corresponding with each of the four diagrams in the display, and, as in the example for step schematics, minor steps consisting of successive actions. While the screen is displayed on the video monitor, they are shown the following text on the computer monitor: "Major Step 1. (MS 1) The seed splits and (MS 2) the hypocotyl emerges (MS 3) to form the primary root. Major Step 2. (MS 4) As the root grows downward (MS 5) the stem breaks through the soil. Major Step 3. (MS 6) The cotyledons open (MS 7) to free the plumule, and (MS 8) the seed coat drops off. Major Step 4. (MS 9) As the stem grows upward, (MS 10) the plumule forms the first leaves." The parts of the schematic in the video screen display highlight when the corresponding text highlights on the computer monitor. After the end of this sequence, students are shown the screen display in FIG. 328 and are told that it is an information matrix <u>classifying</u> the information in each of the four major steps according to whether it represents an agent, action, object, function, direction path, reference point or condition.

<u>Previous Doc</u>   <u>Next Doc</u>   <u>Go to Doc#</u>